

AGENT-BASED AUTOMATED ALGORITHM GENERATOR

Guangfan Zhang, Roger Xu, Xiong Liu, and M. Lyell
Intelligent Automation, Inc.
15400 Calhoun Dr.
Suite 400
Rockville, MD 20855
Telephone: (301) 294-5244
gzhang@i-a-i.com

Xiaodong (Frank) Zhang
Department of Electrical Engineering
Wright State University
3640 Colonel Glenn Highway
Dayton, OH 45435, USA

James Bechtel
US Army RDECOM - TARDEC
6501 E 11 Mile Rd
Building 229, 4th Floor
Warren, MI. 48397

Abstract: The variability of vehicles poses a great challenge on the diagnostics and prognostics for the whole fleet with a vast number of Army ground vehicle platforms. A general diagnostics/prognostics model does not exist and it is difficult to select the best algorithm from a large amount of candidate algorithms for each specific component/subsystem/system application. Therefore, it is necessary to develop a unified framework to evaluate and select the best algorithms, and further maintain the on-vehicle algorithms by updating algorithm parameters and integrating new fleet-wide vehicle data statistics and trends. To address this problem, we propose an agent-based automated algorithm generator for fleet-wide diagnostics/prognostics, which can automatically generate the most suitable algorithm(s) for each vehicle or component in the fleet from a library of light-weight diagnostic/prognostic algorithms. When sufficient fleet-wide statistics and trending information are available, the automated algorithm generator server will automatically determine whether it is necessary to update the current vehicle algorithm configuration or select a better algorithm for on-vehicle diagnostics/prognostics. To prove the concept, we used battery diagnostics as an example to demonstrate the algorithm selection & generation process, and updating capabilities in a networked agent environment.

Key words: Diagnostics; prognostics; health management, automated algorithm generator; agent

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 12 JAN 2010		2. REPORT TYPE N/A		3. DATES COVERED	
4. TITLE AND SUBTITLE Agent-Based Automated Algorithm Generator				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) ; ; ; ; Zhang GuangfanXu RogerLiu XiongLyell M.Zhang XiaodongBechtel James				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Intelligent Automation, Inc. 15400 Calhoun Dr. Suite 400 Rockville, MD 20855 USA Department of Electrical Engineering Wright State University 3640 Colonel Glenn Highway Dayton, OH 45435, USA US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000, USA				8. PERFORMING ORGANIZATION REPORT NUMBER 20577RC	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) TACOM/TARDEC				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 11	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

1. Introduction: A variety of diagnostic and prognostic (D/P) approaches have been developed and implemented for Condition-Based Maintenance of ground vehicles. However, the variability of vehicle's capability, characteristics, and functionality poses a great challenge on the Diagnostics and Prognostics (D/P) for the whole fleet since a universal D/P algorithm does not exist. Instead, for each component/subsystem/system, the most suitable algorithm needs to be assessed and identified individually from a large amount of candidates. Most of current D/P systems have been focusing on how to build a D/P system for a specific platform. Little effort has been made to address the automated D/P algorithm generation. Also, with an algorithm implemented on-vehicle, it is difficult to maintain the algorithm remotely and integrate new fault statistics and trending information.

On the other hand, many existing researches have been focused on addressing D/P capabilities for vehicles and subsystems. In Lee et al's recent paper [2] on the modeling and simulation of vehicle electric power, the battery and generator were modeled and evaluated. They compared two models for battery D/P, and compared a variable terminal voltage alternator model (VTVA) with a constant terminal voltage alternator model (CTVA). In [3], a battery State-Of-Health (SOH) monitoring method was proposed using characteristics of battery voltage signal during vehicle starting. This method doesn't need an expensive high current sensor measuring large battery current signal during engine cranking, and battery SOH is determined based on specific features of the battery voltage signal during engine cranking. A parity relation based approach is introduced in [1] for automotive battery SOH monitoring. A parity relation is designed to characterize the behaviors of good batteries during vehicle cranking. A residual, defined as the discrepancy between the actual battery voltage and its estimation obtained from the trained parity relation, is used to infer battery SOH. Through analysis based on a new battery model during cranking, it is shown that the residual integrates the SOH information provided by battery resistance and voltage loss, hence enhancing D/P performance. B. Saha *et. al.* [4] compared prognostic algorithms for estimating remaining useful life of batteries. Several different methods, including relevance vector machine, support vector machine, and particle filter were compared for estimation of battery remaining useful life (RUL). The performance of these algorithms were compared with classical techniques such as autoregressive integrated moving average and extended Kalman Filtering.

In this paper, we propose a promising solution to fleet-wide D/P, which is an Automated Algorithm Generator (A2G) framework. The A2G contains a library of light-weight D/P algorithms on a server and integrates a server-based toolset that can automatically generate D/P algorithms for each vehicle or component in the fleet. Meanwhile, the A2G server can automatically update and maintain the vehicle D/P algorithms based on the fleet-wide statistics and trending information.

To facilitate the implementation of the A2G, an agent infrastructure is introduced. The Software Requirements Specification (SRS) has been designed for the A2G system, including functional and nonfunctional requirements, such as stakeholder list, actors, metadata structures, interactions, message specifications, hardware/software interface,

and communication protocols. Meanwhile, the whole A2G system is developed under a unified in-house software agent infrastructure. The agent infrastructure is a product for developing Java-based distributed agent applications and provides many useful tools for distributed computing. This allows us to focus on the application level design without having to worry about lower level technical details of the infrastructure. All the services in the A2G system are implemented in the agent infrastructure as autonomous and event-driven agents.

To prove the A2G concept, we have used battery diagnostics as an example to demonstrate the algorithm selection & generation, and updating capabilities in a networked environment.

This paper is organized as follows. Section 2 summarizes the framework of the proposed A2G framework; Section 3 describes the SRS design; Section 4 reports the light-weight D/P algorithms in A2G; Section 5 presents simulation results to prove the A2G concept with an agent framework; and Section 6 gives a summary of the current A2G work and future work.

2. Automated Algorithm Generator (A2G) Framework: The A2G framework contains a library of light-weight algorithms for fleet-wide D/P and has the capabilities of automated algorithm selection, generation, and updating. In this section, the A2G mechanism is introduced followed by algorithm library design.

Figure 1 shows the design of the proposed A2G framework with three functional blocks:

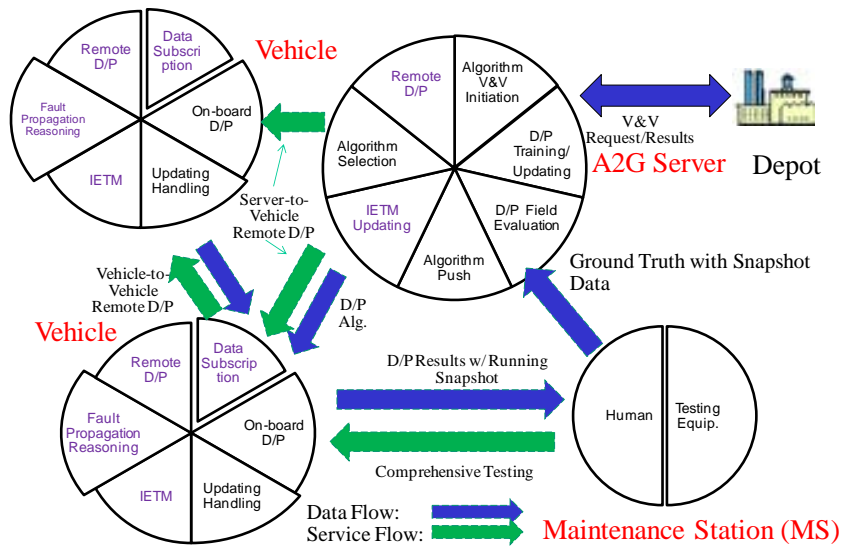


Figure 1 A2G Framework

- **An A2G Server** provides a D/P algorithm library and automated algorithm generation/selection/updating functions.

- **The D/P Engine** onboard each vehicle performs online D/P with the collected signals based on the D/P algorithms generated from the A2G server.
- **A Maintenance Station (MS)** provides a human-in-the-loop environment for evaluating existing D/P results sent from each vehicle, and perform comprehensive testing for a vehicle, if a request is made.

In Figure 1, interconnections among the three functional blocks are illustrated by two different types of arrows: data flow (blue) and/or service flow (green). The data flow shows the communication of data packages among the A2G server, D/P engine onboard each vehicle, and the MS, while the service flow provides a specific D/P service from the A2G server or a MS to a vehicle or even from a vehicle to another vehicle. Different services can be found in the A2G server, vehicle, or MS. The A2G server, for example, includes an algorithm generation and updating service, Interactive Electronic Technical Manual (IETM) updating service, remote D/P service, and field evaluation service.

Automated algorithm generation and updating is a basic function in the A2G framework. The mechanism of automated algorithm generation and updating is shown in Figure 2.

For each type of vehicle or component, who needs the D/P capabilities, the A2G server can automatically select and generate the best algorithms from the algorithm library, following three steps: 1) algorithm query, 2) algorithm selection by training & validation and 3) algorithm selection by depot/OEM level Verification and Validation (V&V). When a

specific type of vehicle or component needs a diagnostic or prognostic algorithm, the algorithm query procedure is first started. By sending a query to a knowledge database containing a library of algorithms, a set of algorithms that meet certain criteria (for example, component type and operating condition) are first selected. The algorithm training/updating service is then utilized to train each selected algorithm, and select a subset of algorithms based on their performance. The selected subset of algorithms will be sent to a depot/OEM for verification and validation and the best performing algorithm will be selected and pushed to the vehicle or component.

After a vehicle receives, installs, and runs the D/P algorithms, the D/P results will be sent to a maintenance station for confirmation. The maintenance station examines the results

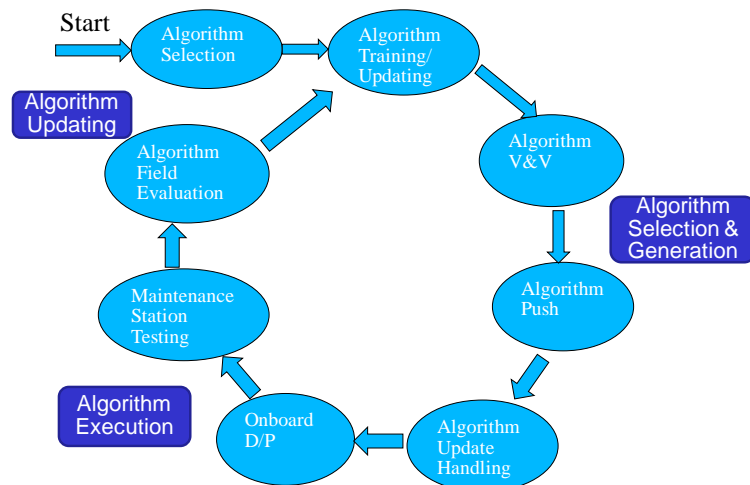


Figure 2 Algorithm generation & updating process

and determines whether the D/P results are correct or accurate and the information is sent back to the A2G server for algorithm evaluation.

Based on the fleet-wide algorithm evaluation, the A2G server determines whether it is necessary to update or reselect algorithms. If necessary, updated or reselected D/P algorithms will be pushed again to the applicable vehicles. In this way, the vehicle-side D/P algorithms have the capabilities of self-adaptation and self-learning utilizing the vehicle and fleet level information.

On a vehicle, the algorithm update handling service is implemented to handle the installation, self-check, and loading of the algorithm being pushed from the server. The onboard D/P service is defined, which takes real-time signals and performs fault detection/identification, and/or prognostics. The D/P results are then sent to a maintenance station for accuracy check and all the D/P results will be reported to the A2G server for field evaluation.

To determine whether it is necessary to update the D/P algorithm, the ground truth or root cause data is collected from the MS and fed into the field evaluation service on the A2G server. If the field evaluation performance of the algorithm, which has been deployed fleet-wide, cannot meet the performance requirement, the algorithm updating process is initiated. The updating is based on both the training data collected previously and the newly added field evaluation data. The performance of the updated D/P algorithm will be compared to that of those algorithms originally trained but not selected due to inferior depot/OEM V&V performance, and the best performing algorithm in the algorithm V&V process will be pushed to the applicable vehicles by the algorithm push service. This way, the vehicle-side D/P has the capabilities of self-adaptation and self-learning utilizing the vehicle and fleet level information.

3. Software Requirements Specification (SRS) Design

To facilitate the implementation of the A2G framework, the SRS needs to be designed, including overall description, external interface requirements, functional requirements, high level design, metadata design, and other nonfunctional requirements. In this section, we briefly explain major items in the SRS design: use cases, functional requirements, and interactions among A2G actors (main role players of the use cases).

In the A2G software, three different use cases are defined:

- Use Case 1: Advanced on-vehicle D/P. This use case generates advanced D/P algorithms and executes them on vehicle. The complexity of the D/P algorithms can be low or medium.
- Use Case 2: Simple on-vehicle D/P. This use case performs simple D/P on vehicle. On the vehicle-side, only simple calculation will be performed, and the D/P will be performed using the simple calculation results using parameters received from the server.

- Use Case 3: On-server D/P. In this use case, complex D/P is performed on server using sensing information received from the vehicle.

These use cases are differentiated based on the following criteria: place holder for D/P algorithms, algorithm complexity, communication burden, and performance, as shown in Table 1.

Table 1 Scenario comparison

Use Case	Real-time Algorithm Place Holder	Complexity	Performance	Communication Burden
Case #1	Vehicle	Medium	Medium/High	Low/Medium
Case #2	Vehicle	Low	Low	Low
Case #3	Server	High	High	High

For each service on the A2G server, vehicle, or Maintenance Station, its main functionality is defined and the required resources to support the service are identified, as well as the communication type used to communicate with other services. Three major communication types are used in the A2G framework: Synchronous Query Response (SQR) that waits for responses, Asynchronous Query Response (AQR) that requests a response that can be sent later, and Subscription (SR) that subscribes to a service. As an example, the following table summarizes the main services on the A2G server.

Table 2 System components, functionality, resources and communication on A2G server

System Components on A2G Server"	Functionality the Component Provides	Resource Needs of the Component	Communication
D/P Field Evaluation	Evaluation of whether the D/P algorithms capture the true health status of vehicles	1. D/P results with ground truth and vehicle data snapshot from maintenance stations 2. Test conditions 3. Original performance expectations	SQR
Algorithm Training/Updating (refinement)	Provides algorithm training and refinement	1. Ground truth data from maintenance stations and vehicles (stored on the server) 2. Performance requirements (False Alarm Rate, <i>etc.</i>) 3. Training conditions	SQR
Algorithm Selection	Determination of the best set of	1. Set of algorithms in algorithm library	AQR

	algorithms and parameters	<ol style="list-style-type: none"> 2. Performance requirements 3. Vehicle specifications 4. Performance evaluation results 	
Remote D/P service	Server-to-vehicle remote D/P	<ol style="list-style-type: none"> 1. Data from the vehicle 2. Vehicle specifications 3. D/P algorithms in use 	AQR
Algorithm V&V Initiation Srv.	Request for algorithm V&V from depots/OEMs	<ol style="list-style-type: none"> 1. Selected D/P algorithms for V&V 2. Expected performance 3. Conditions under which the algorithm is trained 	AQR
IETM Updating	Update IETM	<ol style="list-style-type: none"> 1. Recommended updates 2. Existing IETM 	SQR

For each function, we need to define interactions to achieve the respective functionality. Figure 3 shows the interaction for the algorithm training service that involves different actors: Algorithm Selection Service (ASS), Algorithm Training/Updating Service (ATUS), knowledge database, and algorithm database. The ASS first sends a message to the ATUS with a list of algorithms and training/validation data. The ATUS trains each algorithm using the method provided in the algorithm metadata and performs internal validation on the trained algorithm with the validation dataset. The algorithm metadata will be updated based on the training/validation results and the updated model will be stored in the algorithm database.

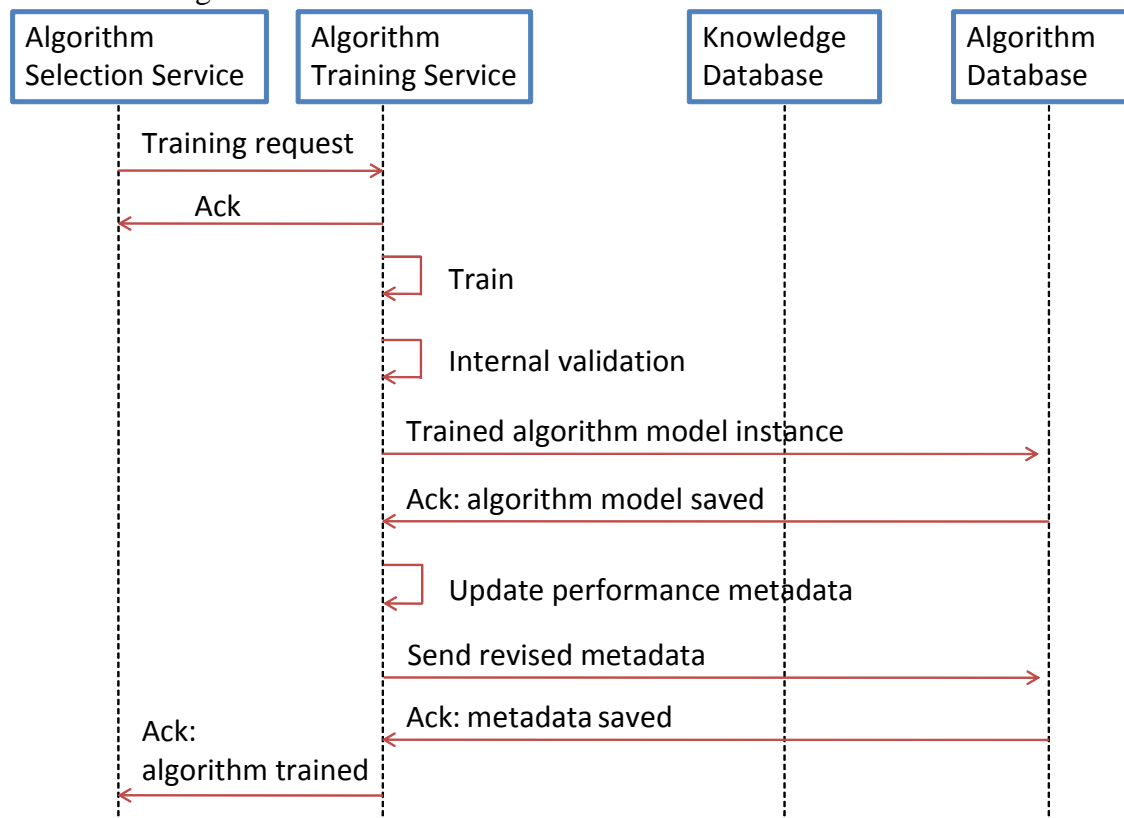


Figure 3 Sequence diagram for algorithm training

4. Light-weight D/P algorithms: To enable the A2G framework for D/P algorithm generation, we need to design and implement a library of light-weight D/P algorithms. In this work, we designed an intelligent multi-agent approach for fleet-wide ground vehicle subsystems/systems D/P. Software agent technology is very useful when developing systems that have many components that have to interact and interoperate with one another (potentially in a concurrent manner and loosely coupled).

The library of D/P algorithms will be hosted in server-side agents, consisting of four types of major agents: Fault Detection and Isolation Agent (FDIA), Prognostic Agent (PA), Fusion Agent (FA), and Maintenance Mining Agent (MMA).

FDI agents perform diagnostics functions utilizing both model-based and data-driven methodologies. Once an anomaly is detected and isolated by the FDI agents, prognostic agents predict the Remaining Useful Life (RUL) with a determined confidence level. Also, maintenance mining agents enables mining of scheduled/unscheduled maintenance data for root-cause analysis and the results can be used by the fusion agents to improve the diagnostic/prognostic performance, such as fault detection robustness, sensitivity, and RUL prediction accuracy and precision. Based on vehicle specifications, system conditions, expert knowledge, historical fault information, test data, and maintenance information, these agents collaborate to build a comprehensive library of D/P models that can be used by individual vehicles.

The D/P model generated by the server will be automatically integrated into a vehicle-side D/P agent to perform health management. No training is required for the vehicle-side D/P agent. A fully functional diagram of the diagnostic/prognostic library can be seen in Figure 4. It is seen that complex modeling, diagnostic/prognostic decision verification and validation, and data mining processes all take place on the server side system. Only light-weight Health Management (HM) agents will be downloaded (migrated) to and executed on the vehicle side system.

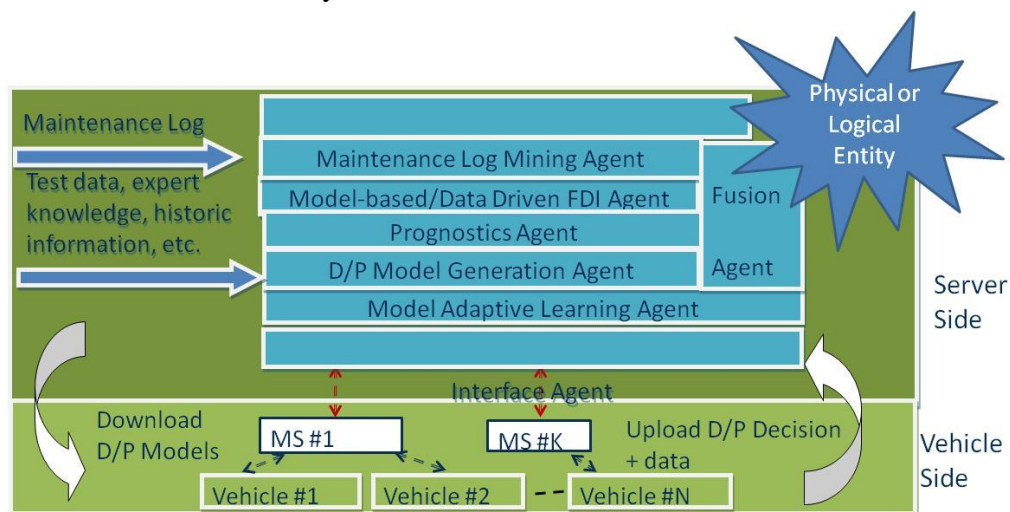


Figure 4 Diagnostic/prognostic library diagram

5. Simulation with Agent Infrastructure: To prove the concept of the A2G framework, we implemented a multi-agent system using the agent infrastructure. Three groups of agents have been developed in A2G, including the server agents, the vehicle agents, and the maintenance station agents. Different agents have been implemented in each group of agents. For example, server side agents include the Algorithm Selection Agent (ASA), the Algorithm Training Agent (ATA), the Model/Algorithm Push Agent (APA), and the Field Evaluation Agent (FEA).

In the simulation, two scenarios are considered: 1) model training and testing using a good battery under 25 °C, and 2) modeling testing and updating using a good battery under -30°C. To show how the A2G works, a diagnostic model is first trained based on the Support Vector Machines method [5]. Two different batteries are first used to train the model under 25°C. Data are collected when the State-Of-Health (SOH) is high and when the SOH is low (before cranking failure). A SVM model is then generated on the A2G server and pushed to the vehicle for onboard D&P. Figure 5 shows a fragment of the model in XML format.

```

1 kobject-stream>
2 <com.rapidminer.operator.ContainerModel id="1">
3   <models id="2">
4     <com.rapidminer.operator.ContainerModel id="3">
5       <models id="4">
6         <com.rapidminer.operator.learner.functions.kernel.LibSVMModel id="5">
7           <model id="6">
8             <param id="7">
9               <svm_type>0</svm_type>
10              <kernel_type>2</kernel_type>
11              <degree>3</degree>
12              <gamma>2.0</gamma>
13              <coef0>0.0</coef0>
14              <cache_size>80.0</cache_size>
15              <eps>0.0010</eps>
16              <C>1.0</C>
17              <nr_weight>2</nr_weight>
18              <weight_label id="8">
19                <int>0</int>
20                <int>1</int>
21              </weight_label>

```

Figure 5 Fragment of the SVM model in XML format

On Board D&P Agent		
● Good battery at 25C ○ Good battery at -30C		
0.96	-0.32	Good
0.96	-0.26	Good
0.98	-0.26	Good
1.00	-0.28	Good
0.99	-0.41	Good
1.07	-0.41	Good
1.03	-0.37	Good
0.95	-0.37	Good
0.95	-0.36	Good
0.94	-0.36	Good
0.96	-0.04	Good
0.95	0.28	Good
0.97	0.48	Good
0.99	0.61	Good
1.04	0.65	Good
1.01	0.76	Good
1.00	1.18	Good

Figure 6 Screenshot where the data of a good battery is used for model testing (under 25 °C)

After training, a new (known good) battery and a known bad (couple of hours before failure) are used to test the model. Figure 6 shows an example when a known good battery is tested onboard the vehicle, and the performance of the onboard diagnostic algorithm is excellent.

However, when a known good battery is used in a different temperature, -30°C, for example, the original onboard diagnostic model will fire many false alarms (confirmed by the maintenance station) since this model has not been trained under this different operating condition. The FEA on the A2G server collects the statistic performance information of the diagnostic algorithm being used and trigger a model updating notification when certain criteria are met (for example, the number of false alarms being detected exceeds a pre-specified threshold).

Therefore, a model updating scheme is needed in this case. We re-trained the model and tested the updated model again, and the results clearly show the performance improvement with over 98% accuracy (as shown in Figure 7). Furthermore, we tested the battery model in -30°C with a battery before a cranking failure, and the diagnostic results found were excellent.

6. Conclusion & Future Work: an A2G framework is developed that can automatically select and generate diagnostic/prognostic algorithms for each vehicle/platform or its components with unsupervised maintenance capability. We have successfully designed the A2G framework with a SRS. Initial simulations and demonstrations have proven the feasibility of the A2G framework using a battery diagnostic example.

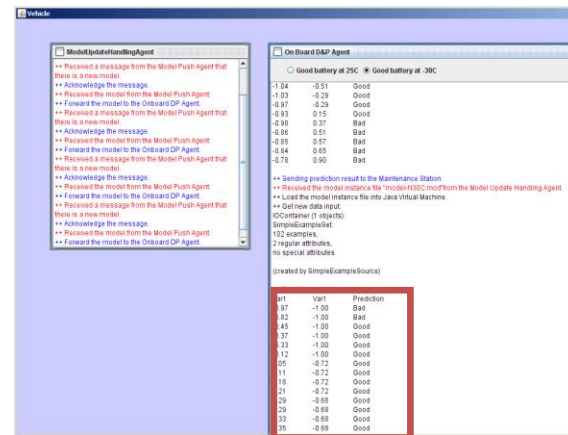


Figure 7 Screenshot where an updated model is installed on the Vehicle Agent Platform

In the current A2G framework, the A2G server has been assumed to be a single entity for simplification. The vehicles in the fleet can be geographically widely located. To guarantee the efficient communication and a more reliable A2G system, possible future work could lead to the development of a hierarchical A2G server structure. The extended framework would contain a centralized A2G server cluster and many local A2G servers. In the centralized A2G server cluster, the functionalities, such as algorithm selection, algorithm push, etc. can be distributed to multiple servers in the cluster for better server management and resource utilization. Meanwhile, multiple local A2G servers are designed to accommodate the spatial distribution of vehicles. A local A2G server is a simplified copy of the central server, without functionalities that rely on fleet-wide information. For example, training service is only designed in the centralized server cluster. In the hierarchical design, the communication to the centralized A2G server cluster can be minimized.

Acknowledgements: This work was supported by Army TARDEC under award W56HZV-09-C-0135. The authors thank Christopher Beck, Ken Fischer, and other Army researchers for valuable comments on this project.

References:

- [1]. X. Zhang, Gruber, M. Salman, and K. Shin, "Automotive battery state-of-health monitoring: a parity relation based approach," the 2009 IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain.
- [2]. W. Lee, D. Choi, M. Sunwoo, "Modeling and simulation of vehicle electric power system," Journal of Power Sources, 109, 2002.

- [3]. X. Zhang, Gruber, M. Salman, K. Shin, "Automotive Battery State-of-Health Monitoring: a Battery Cranking Voltage based Approach," Proc. of the 2008 Integrated System Health Management Conference, Covington, KY.
- [4]. B. Saha, K. Goebel, and J. Christophersen, "Comparison of prognostic algorithms for estimating remaining useful life of batteries," NASA Ames Research Center Technical Report.
- [5]. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>